

**TRƯỜNG ĐẠI HỌC KIÊN GIANG**  
**KHOA THÔNG TIN VÀ TRUYỀN THÔNG**



**TRƯỜNG CƯỜNG THỊNH**

**XÂY DỰNG HỆ THỐNG GIÁM SÁT, CÂN BẰNG NHIỆT  
ĐỘ, ĐỘ ẨM NHÀ YẾN VÀ GỬI CẢNH BÁO BẰNG TIN  
NHẮN ĐIỆN THOẠI**

**KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC**

**Ngành: Công nghệ thông tin**

**Mã số ngành: 7480201**

**Kiên Giang – Năm 2020**

**TRƯỜNG ĐẠI HỌC KIÊN GIANG**  
**KHOA THÔNG TIN VÀ TRUYỀN THÔNG**



**TRƯỜNG CƯỜNG THỊNH**  
**MSSV: 1703206136**

**XÂY DỰNG HỆ THỐNG GIÁM SÁT, CÂN BẰNG NHIỆT  
ĐỘ, ĐỘ ẨM NHÀ YẾN VÀ GỬI CẢNH BÁO BẰNG TIN  
NHẮN ĐIỆN THOẠI**

**KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC**

**Ngành: Công nghệ thông tin**  
**Mã số ngành: 7480201**

**GIẢNG VIÊN HƯỚNG DẪN**  
**Ths. ĐỖ TRUNG KIÊN**

**Kiên Giang – Năm 2020**

## LỜI CẢM ƠN

---

Lời đầu tiên em xin chân thành cảm ơn quý thầy, cô khoa Thông tin và Truyền thông của trường Đại học Kiên Giang đã tận tình dạy bảo và truyền đạt kiến thức cho em trong quá trình học tập và rèn luyện tại Trường, giúp em nắm vững các kiến thức nền tảng để em có thể chuẩn bị tốt và hoàn thành bài báo cáo này.

Em xin chân thành cảm ơn thầy Đỗ Trung Kiên, thầy đã trực tiếp hướng dẫn và góp ý tận tình trong suốt thời gian thực hiện luận văn để em có thể hoàn thành tốt khoá luận này.

Cuối cùng em xin cảm ơn sự chăm sóc của gia đình, sự động viên, giúp đỡ của bạn bè đã tạo điều kiện giúp em hoàn thành tốt khoá luận này.

Mặc dù đã cố gắng hoàn chỉnh các yêu cầu nhưng khoá luận còn một số thiếu sót, mong nhận được sự chỉ bảo, góp ý của quý thầy, cô.

Em xin chân thành cảm ơn!

Châu Thành, ngày.... tháng .... năm 2020

**Sinh viên thực hiện**

**Trương Cường Thịnh**

## **LỜI CAM ĐOAN**

---

Tôi cam đoan rằng khoá luận này là công trình nghiên cứu của bản thân và giáo viên hướng dẫn. Các thông tin về số liệu, hình ảnh, kết quả đã được trình bày trong khoá luận này là trung thực và chưa từng được công bố trong bất kì khoá luận nào trước đây.

Châu Thành, ngày.... tháng .... năm 2020

**Sinh viên thực hiện**

**Trương Cường Thịnh**

[illegible]

**Người hướng dẫn**

## This image shows a full page of white paper designed for handwriting practice. It features approximately 20 evenly spaced horizontal dotted lines running from left to right across the entire width of the page. There are no margins, text, or other markings present.

(Ký và ghi rõ họ tên)

[illegible]

(Ký và ghi rõ họ tên)

## PHIẾU ĐÁNH GIÁ ĐỀ TÀI TỐT NGHIỆP

Họ và tên người đánh giá: .....

Trách nhiệm trong hội đồng: .....

Họ và tên sinh viên: ..... MSSV: .....

Tên đề tài: .....

.....

### Ý KIẾN NHẬN XÉT

#### *1. Đánh giá về chất lượng đề tài tốt nghiệp:*

.....

.....

.....

#### *2. Đánh giá về trình độ, kiến thức của sinh viên* (trình bày và trả lời câu hỏi trước hội đồng)

.....

.....

.....

.....

3. Điểm số (theo thang điểm 10; số lẻ 0,5): .....

4. Ý kiến đề nghị

.....

.....

..... Ngày ... tháng .... năm ....

**Người đánh giá**



## MỤC LỤC

CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN .....	1
1.1. MÔ TẢ BÀI TOÁN.....	1
1.2. MỤC TIÊU CỦA ĐỀ TÀI.....	1
<b>1.2.1. Mục tiêu tổng quát. ....</b>	<b>1</b>
<b>1.2.2. Mục tiêu cụ thể. ....</b>	<b>1</b>
1.3. CÁC CHỨC NĂNG CỦA SẢN PHẨM . ....	2
1.4. MÔI TRƯỜNG VẬN HÀNH. ....	2
<b>1.4.1. Phần cứng. ....</b>	<b>2</b>
<b>1.4.2. Phần mềm. ....</b>	<b>2</b>
1.5. CÁC RÀNG BUỘC THỰC THI VÀ THIẾT KẾ. ....	3
<b>1.5.1. Lập trình nhúng cho board ESP32. ....</b>	<b>3</b>
<b>1.5.2. Hệ thống giám sát theo thời gian thực. ....</b>	<b>3</b>
1.6. HƯỚNG GIẢI QUYẾT VÀ KẾ HOẠCH THỰC HIỆN.....	3
<b>1.6.1. Hướng giải quyết.....</b>	<b>3</b>
<b>1.6.2. Kế hoạch thực hiện. ....</b>	<b>3</b>
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT .....	4
2.1. GIỚI THIỆU VỀ CÔNG NGHỆ INTERNET OF THINGS. ....	4
<b>2.1.2. Hệ thống nhúng (Embedded System). ....</b>	<b>6</b>
<b>2.1.3. Những ứng dụng thực tế .....</b>	<b>6</b>
2.2. Các dịch vụ, nền tảng, giao thức trong hệ thống.....	9
<b>2.2.1. ESP32.....</b>	<b>9</b>
<b>2.2.2. L298N MẠCH CẦU H.....</b>	<b>10</b>
<b>2.2.3. Module Cảm biến DHT11.....</b>	<b>10</b>

2.2.4. NoSQL và hệ quản trị cơ sở dữ liệu MongoDB.....	11
2.2.4.1. NoSQL.....	11
2.2.4.2. Hệ quản trị cơ sở dữ liệu MongoDB.....	12
2.2.5.  Nền tảng Node JS MVC. ....	13
2.2.5.1  Mô hình MVC.....	13
2.2.5.2.  Ngôn ngữ Javascript và nền tảng Node JS. ....	14
2.2.5.3.  Xây dựng ứng dụng với Express JS FrameWork.....	16
2.2.6.  HTML. ....	17
2.2.7.  CSS. ....	18
2.3.  KẾT QUẢ NGHIÊN CỨU LÝ THUYẾT VÀ VẬN DỤNG VÀO ĐỀ TÀI.	19
CHƯƠNG 3: THIẾT KẾ HỆ THỐNG GIÁM SÁT, CÂN BẰNG NHIỆT ĐỘ, ĐỘ ẨM NHÀ YẾN.....	20
3.1.  ĐẶC TẢ HỆ THỐNG. ....	20
3.2.  GIẢI PHÁP THIẾT KẾ HỆ THỐNG .....	20
3.2.1.  Sơ đồ khối hệ thống.....	20
3.2.2.  Nguyên lý hoạt động. ....	21
3.2.2.1.  Tầng vật lý .....	21
3.2.2.2.  Tầng server. ....	22
3.2.2.3.  Tầng client. ....	22
3.2.3.  Lựa chọn linh kiện cho các thiết bị IoT.....	22
3.2.3.1.  Khối xử lý.....	22
3.2.3.2.  Khối cảm biến.....	23
3.2.3.3.  Khối động cơ.....	23
3.2.3.4.  Khối nguồn. ....	25

3.3.	Cài đặt các nền tảng và cấu hình cần thiết.....	25
3.3.1.	Cài đặt Node JS để làm server cho hệ thống.....	25
3.3.2.	Arduino IDE và cấu hình cho chip ESP32 .....	27
3.4.	Thiết kế hệ thống. ....	28
3.4.1.	Thiết kế phần cứng. ....	28
3.4.1.1.	Hệ thống mạch tổng quát.....	28
3.4.1.2.	Hệ thống mạch từng phần.....	29
3.4.2.	Thiết kế cơ sở dữ liệu.....	29
3.4.3.	Thiết kế chức năng cho hệ thống .....	31
3.4.4.	Thiết kế phần mềm. ....	32
3.4.4.1.	Lưu đồ thuật toán của phần lập trình nhúng ESP32. ....	32
3.4.4.2.	Lưu đồ thuật toán phần Server. ....	33
CHƯƠNG 4: KẾT QUẢ - ĐÁNH GIÁ - HƯỚNG PHÁT TRIỂN.....		34
4.1.	Kết quả. ....	34
4.1.1.	Một số hình ảnh hệ thống phần cứng khi hoàn thành. ....	34
4.1.2.	Kết quả từng chức năng.....	34
4.2.	Đánh giá và hướng phát triển .....	40
TÀI LIỆU THAM KHẢO.....		42

## DANH MỤC HÌNH ẢNH

Hình 2.1. Internet of Things (IoT).....	4
Hình 2.2. Sơ đồ thống kê số lượng thiết bị IoT theo thời gian.....	5
Hình 2.3. Hệ thống nhúng (Embedded System).....	6
Hình 2.4. Smart Home.....	7
Hình 2.5. IOT trong nông nghiệp. ....	8
Hình 2.6. IOT đối với sức khỏe con người. ....	8
Hình 2.7. Sơ đồ chân ESP32 .....	9
Hình 2.8. Sơ đồ chân L298N Mạch Cầu H. ....	10
Hình 2.9. Sơ đồ chân module cảm biến DHT11. ....	11
Hình 2.10. NoSQL.....	12
Hình 2.11. Hệ quản trị cơ sở dữ liệu MongoDB .....	13
Hình 2.12. Mô hình MVC. ....	14
Hình 2.13. Ngôn ngữ Javascript. ....	15
Hình 2.14. Nền tảng Node js runtime.....	15
Hình 2.15. Node JS.....	16
Hình 2.16. Cấu trúc của Express JS. ....	16
Hình 2.17. HTML là gì ? .....	17
Hình 2.18. HTML.....	18
Hình 2.19. CSS.....	19
Hình 3.1. Sơ đồ khối hệ thống.....	21
Hình 3.2. Khối xử lý ESP32.....	22
Hình 3.3. Module cảm biến DHT11.....	23
Hình 3.3. Module L298N .....	24

Hình 3.4. Động cơ DC RC520. ....	24
Hình 3.5. Adapter 5V. ....	25
Hình 3.6. Trang chủ Node JS. ....	26
Hình 3.7. Giải nén file Node JS cho Windows. ....	26
Hình 3.8. Kiểm tra version Node JS.....	27
Hình 3.9. Thêm file thông tin cho ESP32. ....	28
Hình 3.10. Hệ thống mạch tổng quát. ....	28
Hình 3.11. Lưu đồ thuật toán phân lập trình nhúng. ....	32
Hình 3.12. Lưu đồ thuật toán phân Server. ....	33
Hình 4.1. Hệ thống phần cứng ....	34
Hình 4.2. Chức năng đăng ký tài khoản. ....	35
Hình 4.3. Chức năng đăng nhập. ....	35
Hình 4.4. Chức năng thêm trạm mới. ....	36
Hình 4.5. Chức năng bản đồ xem nhiều trạm.....	36
Hình 4.6. Chức năng xem thông tin trạm. ....	37
Hình 4.7. Chức năng xem dữ liệu realtime. ....	37
Hình 4.8. Chức năng hiển thị dữ liệu hiện tại ....	38
Hình 4.9. Chức năng thống kê dữ liệu. ....	38
Hình 4.10. Chức năng thống kê số lần vượt ngưỡng. ....	39
Hình 4.11. Chức năng cảnh báo bằng tin nhắn điện thoại. ....	40

# **CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN**

## **1.1. MÔ TẢ BÀI TOÁN.**

Công nghệ cao là một lĩnh vực đang được đầu tư đẩy mạnh phát triển tại Việt Nam và dần chiếm ưu thế trong mọi lĩnh vực, cùng với sự phát triển của thời đại công nghiệp 4.0, đòi hỏi các doanh nghiệp cần ứng dụng công nghệ mới, tiên tiến vào chuyên môn của mình để tăng hiệu quả.

Hiện nay ngành nuôi yến của Việt Nam nói chung, tỉnh Kiên Giang nói riêng đang dần được thịnh hành, đặc biệt là ở khu vực như huyện Châu Thành, thành phố Rạch Giá, để chủ doanh nghiệp hoặc chủ hộ kinh doanh yến có thể dễ dàng kiểm tra các thông số như nhiệt độ, độ ẩm từ xa, thì cần một hệ thống giám sát, từ đó có thể đưa ra những phương pháp phù hợp để cải thiện, phát triển việc kinh doanh.

Và ngày nay, với sự phát triển của Internet, smartphone và đặc biệt là các thiết bị cảm biến, Internet of Things (IoT) đang trở thành xu hướng mới của thế giới. IoT được định nghĩa là những vật dụng có khả năng kết nối Internet. Được ứng dụng trong rất nhiều lĩnh vực khác nhau.

Trong đề tài này, công nghệ IoT sẽ được ứng dụng để xây dựng hệ thống giám sát, cân bằng nhiệt độ, độ ẩm nhà yến và gửi cảnh báo bằng tin nhắn điện thoại. Người dùng có thể giám sát thông số nhiệt độ, độ ẩm qua smartphone hay truy cập vào ứng dụng Web, bất cứ nơi đâu có Internet hoặc WiFi.

## **1.2. MỤC TIÊU CỦA ĐỀ TÀI.**

### **1.2.1. Mục tiêu tổng quát.**

Ứng dụng công nghệ IoT vào ngành công nghiệp công nghệ cao, giúp tăng năng suất mô hình ngành nuôi yến, góp phần phát triển việc ứng dụng công nghệ cao ở Việt Nam.

### **1.2.2. Mục tiêu cụ thể.**

- Xây dựng được hệ thống giám sát chất lượng nhà yến .
- Giúp chủ hộ kinh doanh dễ dàng quản lý các thông số .
- Thống kê, báo cáo, các thông số hàng ngày, hàng tuần .

- Cập nhật thông tin một cách realtime .

### **1.3. CÁC CHỨC NĂNG CỦA SẢN PHẨM .**

Hệ thống giám sát nhiệt độ, độ ẩm theo thời gian thực:

- Đăng nhập, đăng xuất.
- Hiển thị dữ liệu theo thời gian thực.
- Hiển thị thông tin (vị trí, dữ liệu, ngưỡng, cảm biến) của nhiều trạm.
- Thiết lập ngưỡng theo giá trị và màu sắc để cảnh báo trên website.
- Cảnh báo người dùng bằng tin nhắn điện thoại nếu quá ngưỡng.
- Thống kê dữ liệu theo ngày, tuần, tháng.
- Xuất file excel.

### **1.4. MÔI TRƯỜNG VẬN HÀNH.**

#### **1.4.1. Phần cứng.**

- ESP 32 ( main board ).
- Cảm biến DHT11.
- RC520 Động Cơ DC 12VDC + Cánh quạt.
- L298N Mạch Cầu H.
- Dây cắm các loại.
- Nguồn 5V.
- Bảng mạch nguồn.

#### **1.4.2. Phần mềm.**

- Hệ điều hành cho máy tính từ Windows 8 trở lên.
- Môi trường lập trình nhúng cho phần cứng: Arduino IDE.
- Thư viện DHT để đọc chỉ số của cảm biến DHT11.
- CP210x Universal Windows Driver.
- Các thư viện Wifi, ESP để sử dụng board ESP32.
- Node js platform.
- MongoDB đã cấu hình replica set.
- Robo3t để sử dụng MongoDB dễ dàng hơn.
- Visual Studio Code

- WebBrowser: Chrome, Firefox, MS Edge...

## **1.5. CÁC RÀNG BUỘC THỰC THI VÀ THIẾT KẾ.**

### **1.5.1. Lập trình nhúng cho board ESP32.**

- Các thư viện xây dựng sẵn bằng C/C++ phục vụ cho việc đọc, gửi dữ liệu.
- Các thư viện phải có version mới của hiện nay và phải tương thích với main ESP32.
- Lắp các thiết bị cảm biến, module phải đúng chân, đúng nguồn, tránh sai nguồn để gây hư hỏng cho mạch và main board

### **1.5.2. Hệ thống giám sát theo thời gian thực.**

- Hệ thống được phát triển dựa trên ngôn ngữ lập trình Javascript trên nền tảng Node JS
- Sử dụng một số thư viện hỗ trợ cho việc realtime dữ liệu: Ajax, Socket IO.
- Các ngôn ngữ cơ bản để giao diện trở nên thân thiện: Html, Css, Javascript
- Một số thư viện khác: Bootstrap, Leafjs, JQuery ...

## **1.6. HƯỚNG GIẢI QUYẾT VÀ KẾ HOẠCH THỰC HIỆN.**

### **1.6.1. Hướng giải quyết.**

Hệ thống chia thành 3 phần để đảm bảo việc dễ thiết kế và xây dựng:

- Vật lý: các thiết bị vi mạch, các cảm biến gửi thông số đến board mạch chủ ( ESP32 ) thông qua wifi gửi dữ liệu đến server.
- Server : server nhận dữ liệu từ các thiết bị cảm biến thông qua web server và lưu trữ.
- Client : Lấy dữ liệu từ server để render ra cho người dùng thông qua giao diện website.

### **1.6.2. Kế hoạch thực hiện.**

- Nghiên cứu nền tảng, giao thức trong IoT để xây dựng hệ thống.



- Phân tích mô hình tương tác hệ thống giữa tầng vật lý và các tầng trên.
- Phân tích thiết kế hệ thống.
- Xây dựng hệ thống.
- Ứng dụng hệ thống ( chạy thử mô hình demo ).
- Nhận xét và đưa ra hướng phát triển cho sản phẩm trong tương lai.

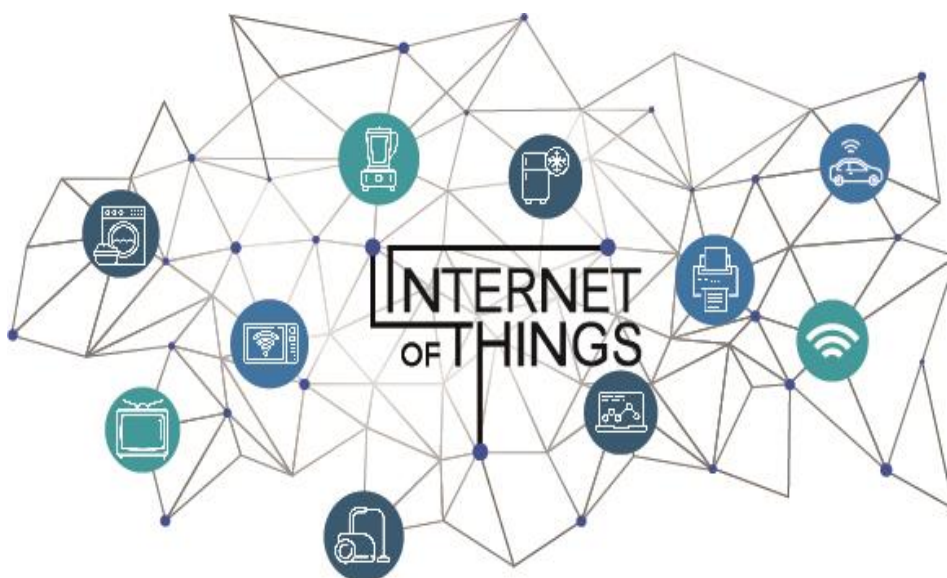
## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## 2.1. GIỚI THIỆU VỀ CÔNG NGHỆ INTERNET OF THINGS.

### 2.1.1. Internet of Things (IoT) là gì ?

Internet of Things, hay IoT, internet vạn vật là đề cập đến hàng tỷ thiết bị vật lý trên khắp thế giới hiện được kết nối với internet, thu thập và chia sẻ dữ liệu. Nhờ bộ xử lý giá rẻ và mạng không dây, có thể biến mọi thứ, từ viên thuốc sang máy bay, thành một phần của IoT. Điều này bổ sung sự “thông minh kỹ thuật số” cho các thiết bị, cho phép chúng giao tiếp mà không cần có con người tham gia và hợp nhất thế giới kỹ thuật số và vật lý.

Internet of things (IoT) dùng để chỉ các đối tượng có thể được nhận biết cũng như chỉ sự tồn tại của chúng trong một kiến trúc tổng hòa mang tính kết nối: Mạng lưới vạn vật kết nối Internet, hay gọi đơn giản hơn là Things.



Hình 2.1. Internet of Things (IoT).

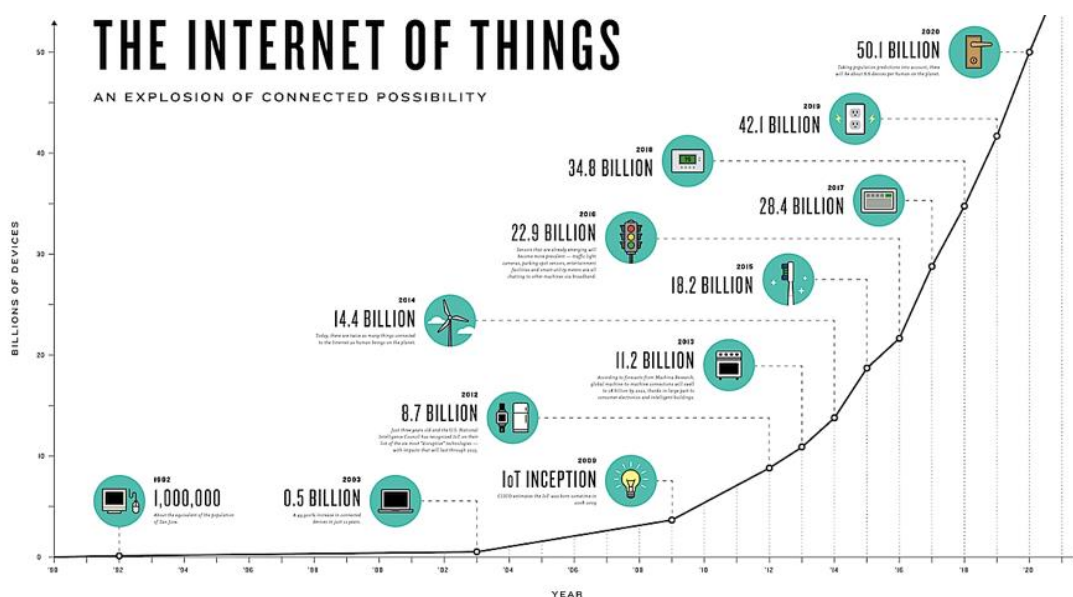
Cụm từ Internet of things được đưa ra bởi Kevin Ashton vào năm 1999, tiếp sau đó nó cũng được dùng nhiều trong các ấn phẩm đến từ các hãng và nhà phân tích. Họ cho rằng IoT là một hệ thống phức tạp, bởi nó là một lượng lớn các đường liên kết giữa máy móc, thiết bị và dịch vụ với nhau.

Ban đầu, IoT không mang ý nghĩa tự động và thông minh. Về sau, người ta đã nghĩ đến khả năng kết hợp giữa hai khái niệm IoT - Autonomous control lại với nhau. Nó có thể quan sát sự thay đổi và phản hồi với môi trường xung quanh, cũng có thể tự điều khiển bản thân mà không cần kết nối mạng. Việc tích hợp trí thông minh vào IoT còn có thể giúp các thiết bị, máy móc, phần mềm thu thập và phân tích các dữ liệu điện tử của con người khi chúng ta tương tác với chúng.

Xu hướng tất yếu trong tương lai, con người có thể giao tiếp với máy móc chỉ qua mạng internet không dây mà không cần thêm bất cứ hình thức trung gian nào khác.

Tình hình trên thế giới hiện nay, tác động của IOT rất đa dạng và tích cực ở nhiều lĩnh vực: quản lý hạ tầng, y tế, xây dựng và tự động hóa, giao thông....

John Chambers (CEO của Cisco) đã công bố: Cho đến năm 2024 sẽ có 500 tỷ thiết bị được kết nối. Thực tế, con số này lớn hơn gần 100 lần số người trên Trái đất, điều đó cho thấy “vạn vật” nhiều hơn con người rất nhiều.



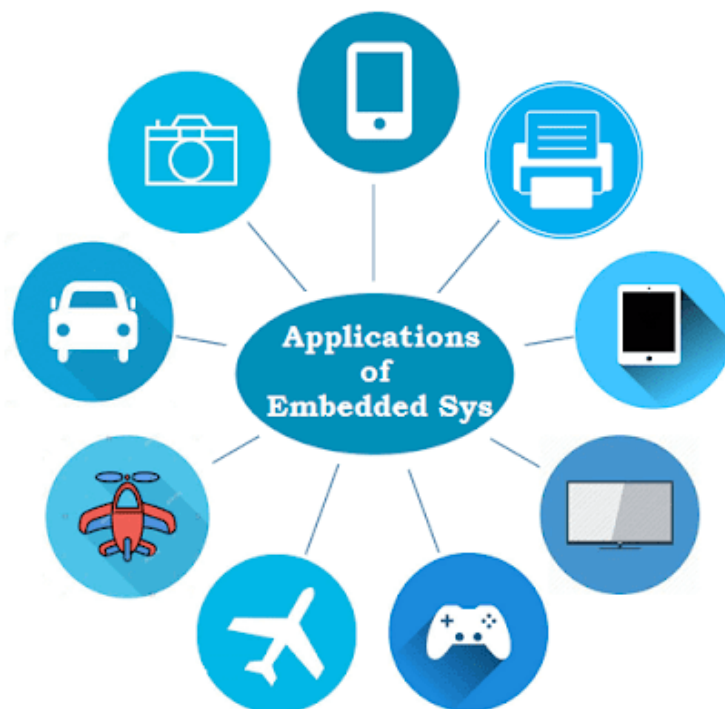
Hình 2.2. Sơ đồ thống kê số lượng thiết bị IoT theo thời gian.

Người ta cho rằng, IoT là chìa khóa của sự thành công, là bước ngoặt và cơ hội lớn của tương lai. Để không bị tụt lại phía sau, các chính phủ và doanh nghiệp

cần có sự đổi mới và đầu tư mạnh tay hơn để phát triển các sản phẩm ứng dụng công nghệ Internet of things.

### 2.1.2. Hệ thống nhúng (Embedded System).

Hệ thống nhúng (embedded system): là một thuật ngữ để chỉ một hệ thống có khả năng tự trị được nhúng vào trong một môi trường hay hệ thống mẹ. Đó là các hệ thống tích hợp cả phần cứng và phần mềm phục vụ các bài toán chuyên dụng trong nhiều lĩnh vực công nghiệp, tự động hoá điều khiển, quan trắc và truyền tin. Đặc điểm của các hệ thống nhúng là hoạt động ổn định và có tính năng tự động hoá cao.



Hình 2.3. Hệ thống nhúng (Embedded System)

### 2.1.3. Những ứng dụng thực tế

**Smart Home:** Đối với người tiêu dùng, ngôi nhà thông minh có lẽ là nơi họ có khả năng tiếp xúc với những thứ có kết nối internet và đó là một lĩnh vực mà các công ty công nghệ lớn (đặc biệt là Amazon, Google và Apple) đang cạnh tranh gay gắt.

Rõ ràng nhất trong số này là các smart speakers như Echo của Amazon, nhưng cũng có phích cắm thông minh, bóng đèn, máy ảnh, máy điều nhiệt và tủ

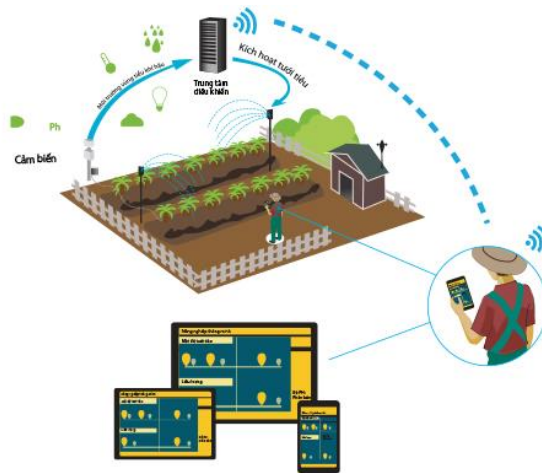
lạnh thông minh bị đánh giá không tốt. Chúng có thể giúp người già độc lập và ở nhà lâu hơn bằng cách giúp gia đình và người chăm sóc dễ dàng giao tiếp và theo dõi. Ngoài ra có thể giúp tiết kiệm năng lượng chẳng hạn như cắt giảm chi phí sưởi ấm.



*Hình 2.4. Smart Home.*

**IoT trong nông nghiệp:** Với sự gia tăng liên tục của dân số đồng nghĩa với việc nhu cầu sử dụng lương thực tăng lên nhiều lần. Nông dân có thể áp dụng các kỹ thuật mới, công nghệ tiên tiến để tăng sản lượng sản xuất nông nghiệp. Nông nghiệp thông minh có thể nói là lĩnh vực phát triển nhanh nhất với **IoT**.

Những thông tin người nông dân thu được giúp họ có những quyết định đầu tư sáng suốt tránh tình trạng “được mùa mất giá, được giá mất mùa” như hiện nay. Cảm biến độ ẩm, chất dinh dưỡng của đất, mức độ hấp thụ nước góp phần quan trọng vào việc kiểm soát sự tăng trưởng của cây trồng giúp người gieo trồng có thể xác định, tùy chỉnh lượng phân bón cần thiết.



*Hình 2.5. IOT trong nông nghiệp.*

**Sức khỏe:** Đây có thể nói là một lĩnh vực chưa được khai phá hết của Internet of Things bởi những ứng dụng không ngờ mà nó mang lại. Một hệ thống chăm sóc sức khỏe được kết nối cùng các thiết bị y tế thông minh mang lại tiềm năng to lớn cho các công ty đầu tư sản xuất. IoT trong chăm sóc sức khỏe giúp mọi người có cuộc sống khỏe mạnh hơn bằng việc đeo các thiết bị kết nối. Các dữ liệu thu thập được giúp phân tích sức khỏe của người dùng thiết bị kết nối và nhà cung cấp, sản xuất sẽ có được những thiết kế để chống lại bệnh tật.



*Hình 2.6. IOT đối với sức khỏe con người.*



## 2.2. Các dịch vụ, nền tảng, giao thức trong hệ thống.

### 2.2.1. ESP32.

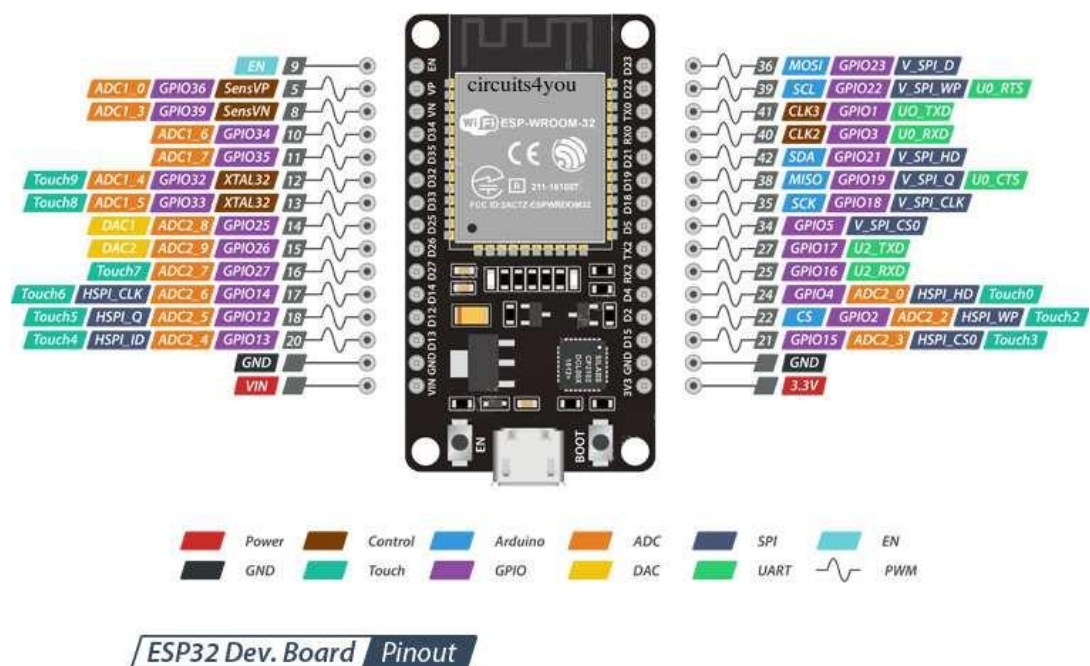
ESP32 là dòng chip tích hợp Wi-Fi 2.4Ghz có thể lập trình được, rẽ tiền được sản xuất bởi một công ty bán dẫn Trung Quốc: Espressif Systems.

Được đóng gói đưa ra thị trường dạng Module, được sản xuất bởi bên thứ 3: AI-Thinker. Có khả năng kết nối Internet qua mạng Wi-Fi một cách nhanh chóng và sử dụng rất ít linh kiện đi kèm. Với giá cả có thể nói là rất rẻ so với tính năng và khả năng ESP32 có thể làm được.

ESP32 có một cộng đồng các nhà phát triển trên thế giới rất lớn, cung cấp nhiều Module lập trình mã nguồn mở giúp nhiều người có thể tiếp cận và xây dựng ứng dụng rất nhanh.

SDK hỗ trợ chính thức từ hãng.

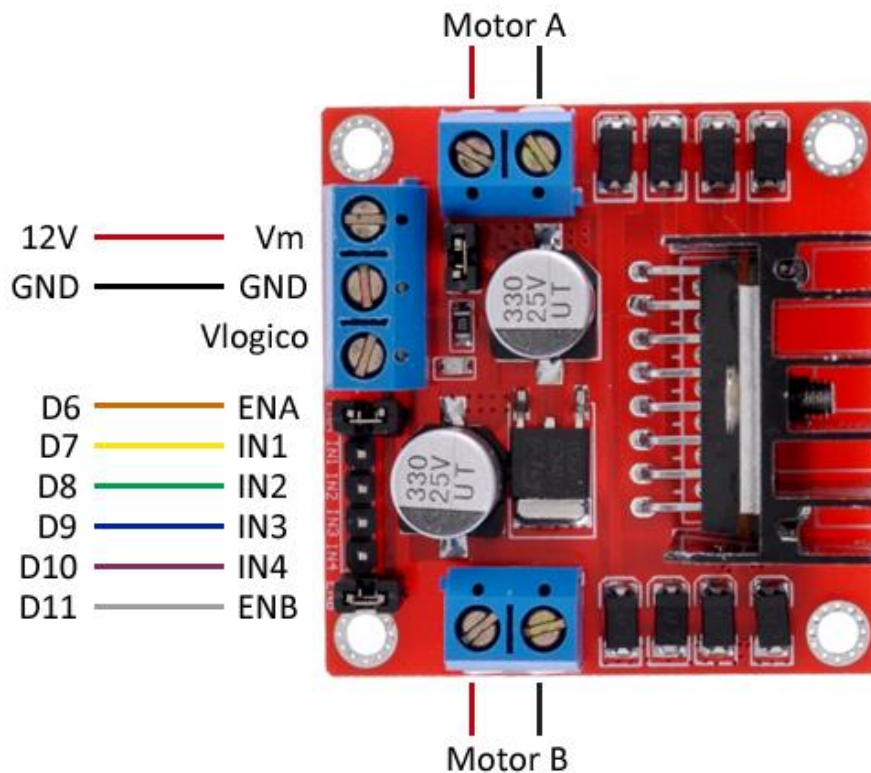
Espressif hiện đã hỗ trợ 3 nền tảng SDK (Software Development Kit - Gói phát triển phần mềm) độc lập, là: NONOS SDK, RTOS SDK và Arduino. Cả 3 đều có những ưu điểm riêng phù hợp với từng ứng dụng nhất định, và sử dụng chung nhiều các hàm điều khiển phần cứng. Hiện nay Arduino đang được sử dụng rộng rãi bởi tính dễ sử dụng, kiến trúc phần mềm tốt và tận dụng được nhiều thư viện cộng đồng.



Hình 2.7. Sơ đồ chân ESP32

### 2.2.2. L298N MẠCH CẦU H.

Module điều khiển động cơ (Motor Driver) sử dụng chip cầu H L298N giúp điều khiển tốc độ và chiều quay của động cơ DC một cách dễ dàng, ngoài ra module L298N còn điều khiển được 1 động cơ bước lưỡng cực. Mạch cầu H của IC L298N có thể hoạt động ở điện áp từ 5V đến 35V.

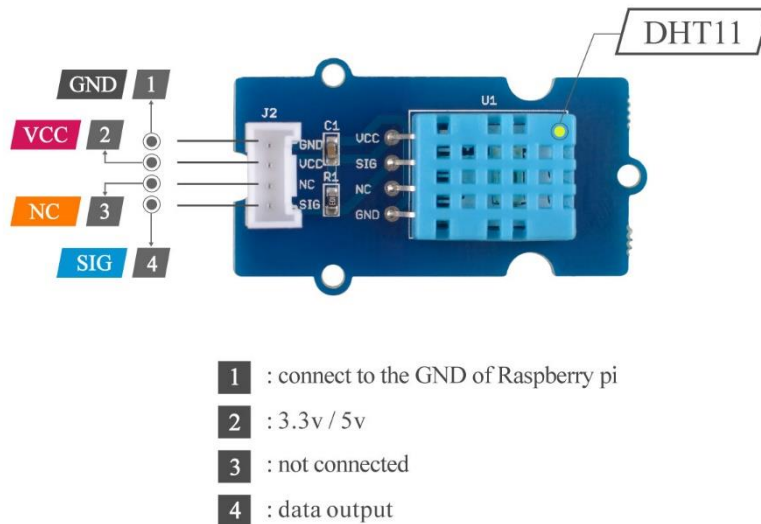


Hình 2.8. Sơ đồ chân L298N Mạch Cầu H.

### 2.2.3. Module Cảm biến DHT11.

Cảm biến độ ẩm và nhiệt độ DHT11 là cảm biến rất thông dụng hiện nay vì chi phí rẻ và rất dễ lấy dữ liệu thông qua chuẩn giao tiếp 1 wire. Chuẩn giao tiếp 1 wire là dùng 1 chân Digital để truyền dữ liệu.

Bộ tiền xử lý tín hiệu được tích hợp trong cảm biến giúp bạn có thể đọc dữ liệu chính xác mà không phải qua bất kỳ tính toán nào.



Hình 2.9. Sơ đồ chân module cảm biến DHT11.

## 2.2.4. NoSQL và hệ quản trị cơ sở dữ liệu MongoDB.

### 2.2.4.1. NoSQL.

- NoSQL là 1 dạng CSDL mã nguồn mở và được viết tắt bởi: None-Relational SQL hay có nơi thường gọi là Not-Only SQL.
- NoSQL được phát triển trên Javascript Framework với kiểu dữ liệu là JSON và dạng dữ liệu theo kiểu key và value.
- NoSQL ra đời như là 1 mảnh vá cho những khuyết điểm và thiếu sót cũng như hạn chế của mô hình dữ liệu quan hệ RDBMS (Relational Database Management System - Hệ quản trị cơ sở dữ liệu quan hệ) về tốc độ, tính năng, khả năng mở rộng,...
- Với NoSQL bạn có thể mở rộng dữ liệu mà không lo tới những việc như tạo khóa ngoại, khóa chính, kiểm tra ràng buộc .v.v ...
- NoSQL bỏ qua tính toàn vẹn của dữ liệu và transaction để đổi lấy hiệu suất nhanh và khả năng mở rộng.
- NoSQL được sử dụng ở rất nhiều công ty, tập đoàn lớn, ví dụ như FaceBook sử dụng Cassandra do FaceBook phát triển, Google phát triển và sử dụng BigTable,...





## Non Structured Query Language

*Hình 2.10. NoSQL.*

### 2.2.4.2. Hệ quản trị cơ sở dữ liệu MongoDB.

- MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở, là CSDL thuộc NoSql và được hàng triệu người sử dụng.
- MongoDB là một database hướng tài liệu (document), các dữ liệu được lưu trữ trong document kiểu JSON thay vì dạng bảng như CSDL quan hệ nên truy vấn sẽ rất nhanh.
- Với CSDL quan hệ chúng ta có khái niệm bảng, các cơ sở dữ liệu quan hệ (như MySQL hay SQL Server...) sử dụng các bảng để lưu dữ liệu thì với MongoDB chúng ta sẽ dùng khái niệm là **collection** thay vì bảng

- So với RDBMS thì trong MongoDB **collection** ứng với **table**, còn **document** sẽ ứng với **row**, MongoDB sẽ dùng các document thay cho row trong RDBMS.
- Các collection trong MongoDB được cấu trúc rất linh hoạt, cho phép các dữ liệu lưu trữ không cần tuân theo một cấu trúc nhất định.
- Thông tin liên quan được lưu trữ cùng nhau để truy cập truy vấn nhanh thông qua ngôn ngữ truy vấn MongoDB



*Hình 2.11. Hệ quản trị cơ sở dữ liệu MongoDB*

## 2.2.5. Nền tảng Node JS MVC.

### 2.2.5.1 Mô hình MVC.

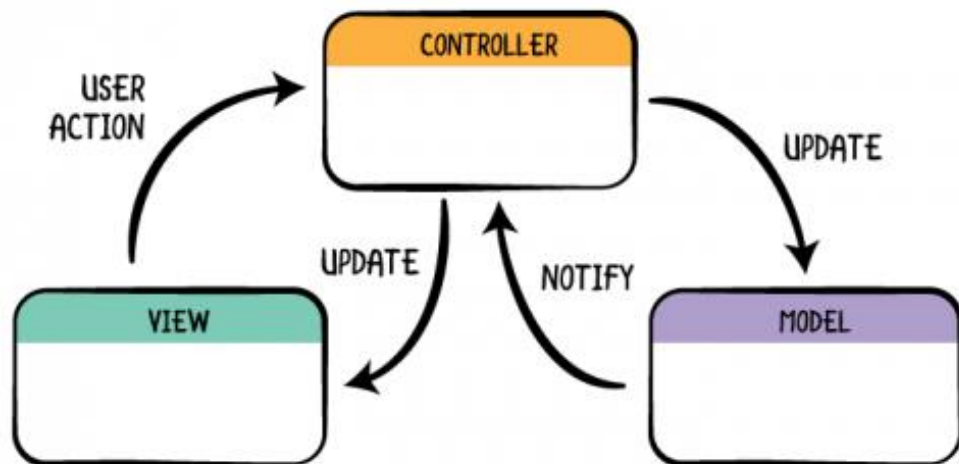
MVC là từ viết tắt của '**Model View Controller**'. Nó đại diện cho các nhà phát triển kiến trúc áp dụng khi xây dựng các ứng dụng. Với kiến trúc MVC, chúng ta xem xét cấu trúc ứng dụng liên quan đến cách luồng dữ liệu của ứng dụng của chúng ta hoạt động như thế nào.

#### **Các thành phần trong mô hình MVC:**

Mô hình MVC được chia làm 3 lớp xử lý gồm Model – View – Controller :

- **Model** : là nơi chứa những nghiệp vụ tương tác với dữ liệu hoặc hệ quản trị cơ sở dữ liệu (mysql, mssql... ); nó sẽ bao gồm các class/function xử lý nhiều nghiệp vụ như kết nối database, truy vấn dữ liệu, thêm – xóa – sửa dữ liệu...

- **View** : là nơi chứa những giao diện như một nút bấm, khung nhập, menu, hình ảnh... nó đảm nhiệm nhiệm vụ hiển thị dữ liệu và giúp người dùng tương tác với hệ thống.
- **Controller** : là nơi tiếp nhận những yêu cầu xử lý được gửi từ người dùng, nó sẽ gồm những class/ function xử lý nhiều nghiệp vụ logic giúp lấy đúng dữ liệu thông tin cần thiết nhờ các nghiệp vụ lớp Model cung cấp và hiển thị dữ liệu đó ra cho người dùng nhờ lớp View.



Hình 2.12. Mô hình MVC.

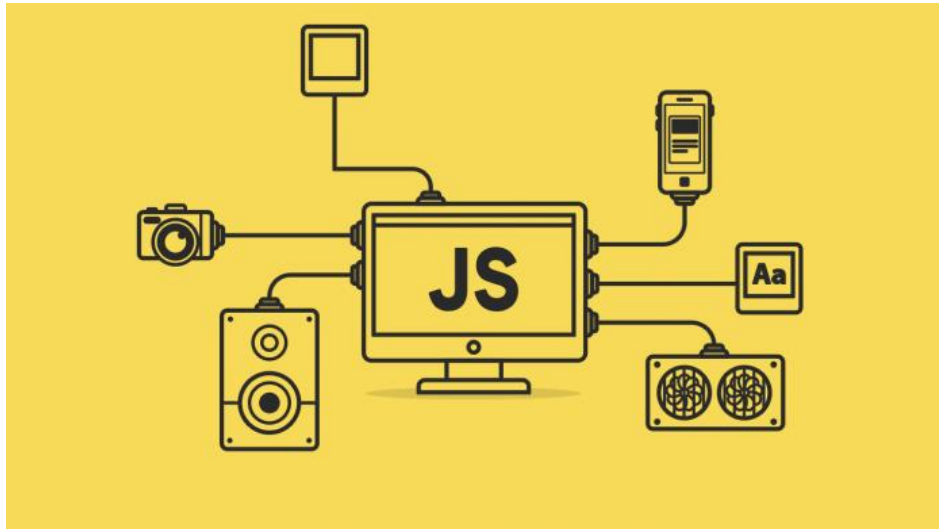
#### Sự tương tác giữa các thành phần:

- **Controller** tương tác với qua lại với **View**
- **Controller** tương tác qua lại với **Model**
- **Model** và **View** không có sự tương tác với nhau mà nó tương tác với nhau thông qua **Controller**.

#### 2.2.5.2. Ngôn ngữ Javascript và nền tảng Node JS.

##### Ngôn ngữ kịch bản Javascript

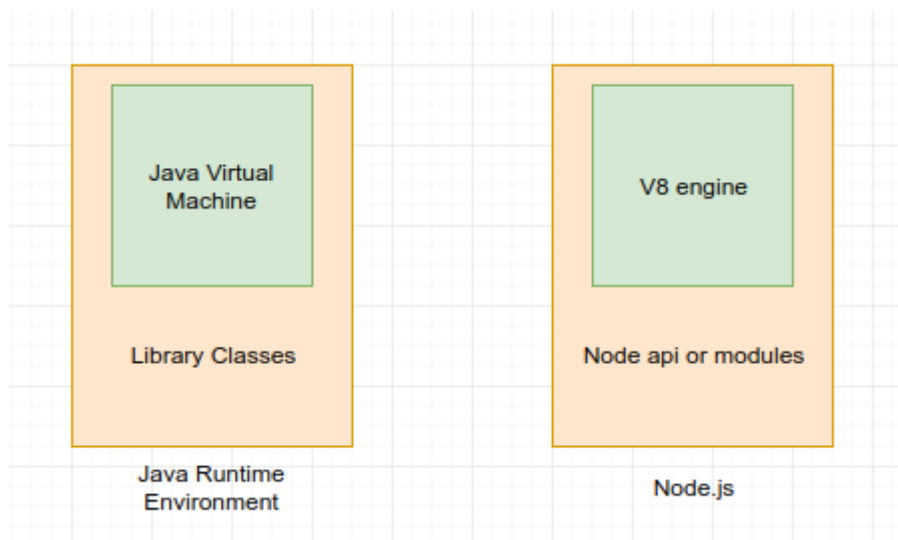
- JavaScript là ngôn ngữ lập trình phổ biến nhất trên thế giới trong suốt 20 năm qua. Nó cũng là một trong số 3 ngôn ngữ chính của lập trình web: HTML, CSS, JavaScript –
- JavaScript, theo phiên bản hiện hành, là một ngôn ngữ lập trình thông dịch được phát triển từ các ý niệm nguyên mẫu. Ngôn ngữ này được dùng rộng rãi cho các trang web cũng như phía máy chủ.



*Hình 2.13. Ngôn ngữ Javascript.*

### Nền tảng Node JS

- Nền tảng Node runtime bao gồm mọi thứ mà bạn cần để hoàn thành một chương trình viết bằng JavaScript.



*Hình 2.14. Nền tảng Node js runtime.*

- **NodeJS** là một nền tảng được xây dựng trên “V8 Javascript engine” được viết bằng c++ và Javascript. Nền tảng này được phát triển bởi Ryan Lienhart Dahl vào năm 2009.
- **NodeJS** ra đời khi các developer đòi đầu của JavaScript mở rộng nó từ một thứ bạn chỉ chạy được trên trình duyệt thành một thứ bạn có thể chạy trên máy của mình dưới dạng ứng dụng độc lập.
- Cả trình duyệt JavaScript và Node.js đều chạy trên JavaScript runtime V8 engine. Công cụ này lấy code JavaScript của bạn và convert nó sang mã

máy (bytecode) cho việc thực thi nhanh hơn. Mã máy là loại code thấp cấp hơn để máy tính có thể chạy mà không cần biên dịch nó.

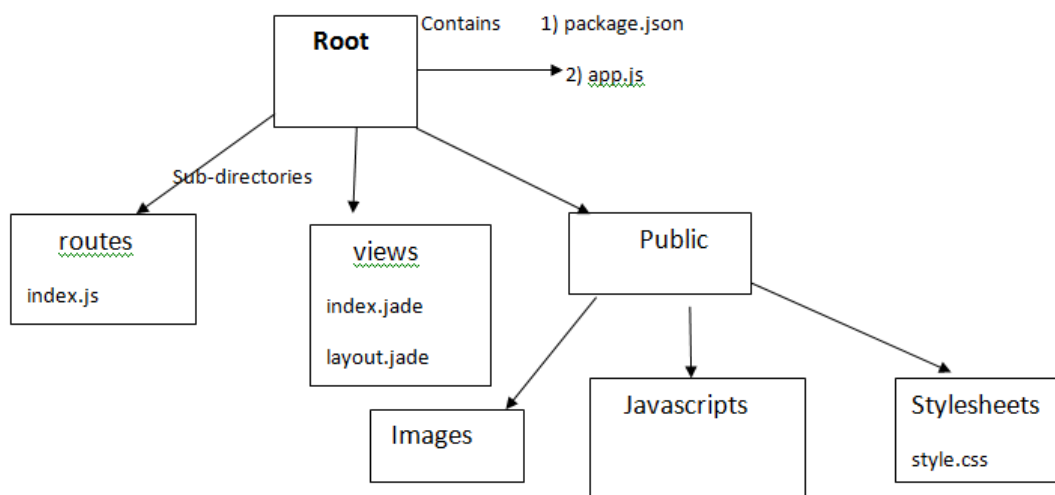


Hình 2.15. Node JS.

### 2.2.5.3. Xây dựng ứng dụng với Express JS Framework

- Express js là một Framework nhỏ, nhưng linh hoạt được xây dựng trên nền tảng của Nodejs. Nó cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile
  - Về các package hỗ trợ: Expressjs có vô số các package hỗ trợ nên các bạn không phải lo lắng khi làm việc với Framework này.
  - Về performance: Express cung cấp thêm về các tính năng (feature) để dev lập trình tốt hơn. Chứ không làm giảm tốc độ của NodeJS.
  - Và hơn hết, các Framework nổi tiếng của NodeJS hiện nay đều sử dụng ExpressJS như một core function, chẳng hạn: SailsJS, MEAN,....
- Các bạn tham khảo ở đây

#### Cấu trúc của Express JS:



Hình 2.16. Cấu trúc của Express JS.

### 2.2.6. HTML.



*Hình 2.17. HTML là gì ?*

HTML viết tắt của Hypertext Markup Language là ngôn ngữ lập trình dùng để xây dựng và cấu trúc lại các thành phần có trong Website.

HTML tạm dịch là ngôn ngữ đánh dấu siêu văn bản. Người ta thường sử dụng HTML trong việc phân chia các đoạn văn, heading, links, blockquotes,...

HTML được sáng tạo bởi **Tim Berners-Lee**, nhà vật lý học của trung tâm nghiên cứu CERN ở Thụy Sĩ. Anh ta đã nghĩ ra được ý tưởng cho hệ thống hypertext trên nền Internet.

Hypertext có nghĩa là văn bản chứa links, nơi người xem có thể truy cập ngay lập tức. Anh xuất bản phiên bản đầu tiên của HTML trong năm 1991 bao gồm 18 tag HTML. Từ đó, mỗi phiên bản mới của HTML đều có thêm tag mới và attributes mới.

Theo Mozilla Developer Network: HTML Element Reference, hiện tại có hơn 140 HTML tags, mặc dù một vài trong số chúng đã bị tạm ngưng (không hỗ trợ bởi các trình duyệt hiện đại).

Nhanh chóng phổ biến ở mức độ chóng mặt, HTML được xem như là chuẩn mực của một website. Các thiết lập và cấu trúc HTML được vận hành và phát triển bởi World Wide Web Consortium (W3C). Bạn có thể kiểm tra tình trạng mới nhất của ngôn ngữ này bất kỳ lúc nào trên trang **W3C's website**.

Nâng cấp mới nhất gần đây là vào năm 2014, khi ra mắt chuẩn HTML5. Nó thêm vài tags vào markup, để xác định rõ nội dung thuộc loại là gì, như là <article>, <header>, và <footer>.



Hình 2.18. HTML.

Cấu trúc của 1 trang HTML gồm:

1. Tag **<html></html>** là element cao nhất dùng để đóng gói mỗi trang HTML.
2. Tag **<head></head>** chứa các thông tin meta như là tiêu đề trang và charset.
3. Cuối cùng, **<body></body>** tag dùng để đóng gói tất cả nội dung sẽ hiện trên trang.

#### 2.2.7. CSS.

- CSS là viết tắt của Cascading Style Sheets. Đây là một ngôn style sheet được sử dụng để mô tả giao diện và định dạng của một tài liệu viết bằng ngôn ngữ đánh dấu (markup). Nó cung cấp một tính năng bổ sung cho HTML. Nó thường được sử dụng với HTML để thay đổi phong cách của trang web và giao diện người dùng. Nó cũng có thể được sử dụng với bất kỳ loại tài liệu XML nào bao gồm cả XML đơn giản, SVG và XUL.



*Hình 2.19. CSS.*

- CSS được sử dụng cùng với HTML và JavaScript trong hầu hết các trang web để tạo giao diện người dùng cho các ứng dụng web và giao diện người dùng cho nhiều ứng dụng di động.

### **2.3. KẾT QUẢ NGHIÊN CỨU LÝ THUYẾT VÀ VẬN DỤNG VÀO ĐỀ TÀI.**

Hệ thống đo nhiệt độ, độ ẩm ứng dụng công nghệ IoT cho nhà yến đã thực hiện các công việc sau:

- Lắp đặt các thiết bị phần cứng kết nối với main board ESP32.
- Kết nối ESP32 với wifi để gửi dữ liệu đã đọc từ các cảm biến.
- Webserver nhận dữ liệu từ ESP32 lưu vào DB.
- Webserver nhận biết collection dữ liệu thay đổi sau đó lấy dữ liệu mới nhất để gửi về view thông qua socket io.
- Client nhận dữ liệu mới nhất của Webserver để render ra cho người dùng theo thời gian thực.



## **CHƯƠNG 3: THIẾT KẾ HỆ THỐNG GIÁM SÁT, CÂN BẰNG NHIỆT ĐỘ, ĐỘ ẨM NHÀ YẾN**

### **3.1. ĐẶC TẢ HỆ THỐNG.**

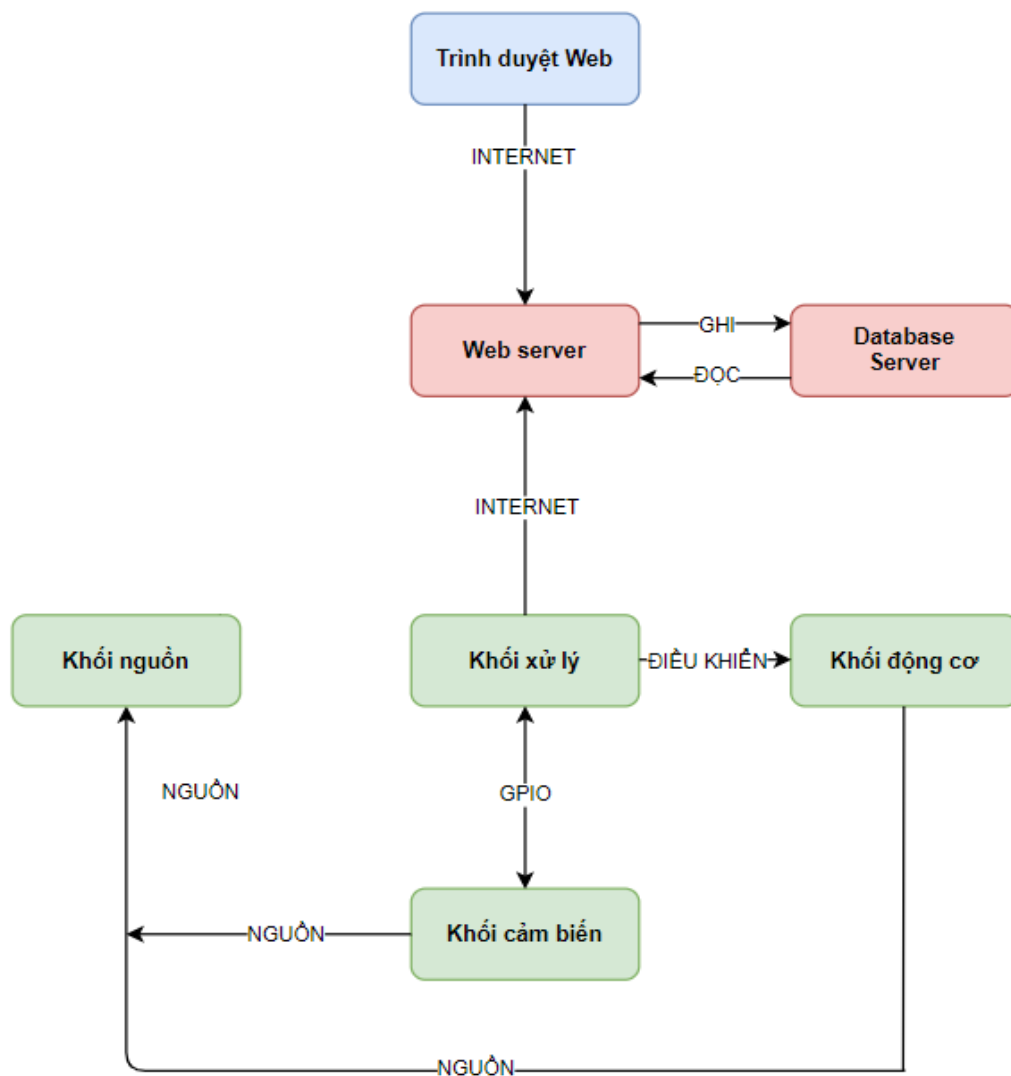
- Hệ thống ứng dụng công nghệ IoT để giám sát, cân bằng nhiệt độ, độ ẩm và gửi cảnh báo bằng tin nhắn điện thoại là hệ thống kết hợp giữa phần cứng và phần mềm .
- Hệ thống phần cứng chủ yếu do board ESP32 điều khiển thông qua wifi, internet để giao tiếp với cơ sở dữ liệu và webserver.
- Main board ESP32 đọc thông số nhiệt độ, độ ẩm của module DHT11 gửi lên cho webserver.
- Webserver xử lý dữ liệu và trả về cho giao diện người dùng dưới dạng bảng, biểu đồ theo thời gian thực, nếu quá ngưỡng và đủ số mẫu lấy dữ liệu thì sẽ cảnh báo bằng màu sắc trên giao diện website, thêm vào đó sẽ cảnh báo thêm cho người dùng bằng tin nhắn điện thoại.

### **3.2. GIẢI PHÁP THIẾT KẾ HỆ THỐNG**

#### **3.2.1. Sơ đồ khối hệ thống.**

Sơ đồ khối gồm 3 tầng:

1. Tầng vật lý: khối xử lý, khối nguồn, khối cảm biến, khối điều khiển.
2. Tầng server: web server, database server.
3. Tầng client: trình duyệt web.



Hình 3.1. Sơ đồ khối hệ thống

### 3.2.2. Nguyên lý hoạt động.

#### 3.2.2.1. Tầng vật lý

- Khối cảm biến: module cảm biến DHT11 sau khi đọc cảm chỉ số nhiệt độ, độ ẩm thông qua thư viện DHT.
- Khối nguồn: cung cấp nguồn cho tất cả hệ thống ở tầng vật lý.
- Khối xử lý: nhận dữ liệu từ DHT11 điều khiển khối động cơ hoạt động nếu quá ngưỡng đã cài đặt, gửi tín hiệu cho webserver thông qua internet.

- Khối động cơ: hoạt động dựa sự điều khiển của khối xử lý, nếu quá ngưỡng thì quay các motor đã được cài đặt trong hệ thống.

### 3.2.2.2. Tầng server.

- Web server: nhận tín hiệu từ khối xử lý ở tầng vật lý thông qua wifi, internet và lưu vào database server khi đã thông qua các kiểm tra trên code. Đọc dữ liệu và gửi cho trình duyệt web ở tầng client.
- Database server: lưu trữ tất cả dữ liệu của toàn bộ hệ thống với hệ quản trị cơ sở dữ liệu.

### 3.2.2.3. Tầng client.

- Trình duyệt web: nhận dữ liệu từ webserver và hiển thị ở dạng biểu đồ, dạng bảng, cảnh báo bằng màu sắc khi vượt quá ngưỡng, hiển thị thông tin của các trạm .

## 3.2.3. Lựa chọn linh kiện cho các thiết bị IoT

### 3.2.3.1. Khối xử lý.

Khối xử lý trung tâm được đảm nhiệm bởi chip ESP32 của AIThinker.



Hình 3.2. Khối xử lý ESP32

Với hiệu suất cao, bộ xử lý mạnh mẽ, hỗ trợ các chế độ: AP, STA, và AP+STA, module wifi ESP32 dễ dàng xử lý các logic của chương trình thông qua wifi để hệ thống có thể hoạt động theo thời gian thực.

### 3.2.3.2. Khối cảm biến.

Khối cảm biến sử dụng module cảm biến nhiệt độ, độ ẩm DHT11.



Hình 3.3. Module cảm biến DHT11.

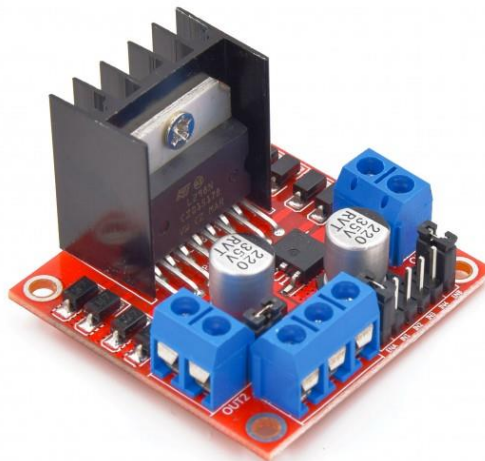
- Điện áp hoạt động: 5VDC
- Chuẩn giao tiếp: TTL, 1 wire.
- Khoảng đo độ ẩm: 20%-80%RH sai số  $\pm 5\%$  RH
- Khoảng đo nhiệt độ: 0-50°C sai số  $\pm 2^\circ\text{C}$
- Tần số lấy mẫu tối đa 1Hz (1 giây / lần)
- Kích thước: 28mm x 12mm x 10mm

Module có thể đo được cả 2 thông số nhiệt độ và độ ẩm như các họ cảm biến DHT khác như DHT22 ..., với độ sai số không quá lớn module cảm biến có thể cung cấp các thông số gần đúng so với môi trường xung quanh.

### 3.2.3.3. Khối động cơ.

Khối động cơ gồm 2 phần chính :

**Module L298N Mạch cầu H:**



*Hình 3.3. Module L298N*

- IC chính: L298 – Dual Full Bridge Driver
- Điện áp đầu vào: 5~30VDC
- Công suất tối đa: 25W 1 cầu (lưu ý công suất = dòng điện x điện áp nên áp cấp vào càng cao, dòng càng nhỏ, công suất có định 25W).
- Dòng tối đa cho mỗi cầu H là: 2A
- Mức điện áp logic: Low -0.3V~1.5V, High: 2.3V~Vss
- Kích thước: 43x43x27mm

**Động cơ DC RC520:**



*Hình 3.4. Động cơ DC RC520.*

- Động cơ DC 12V
- Tốc độ: 6000RPM
- Size 22 x 33MM.

#### **3.2.3.4. Khôi nguồn.**

Sử dụng nguồn adapter 5V:



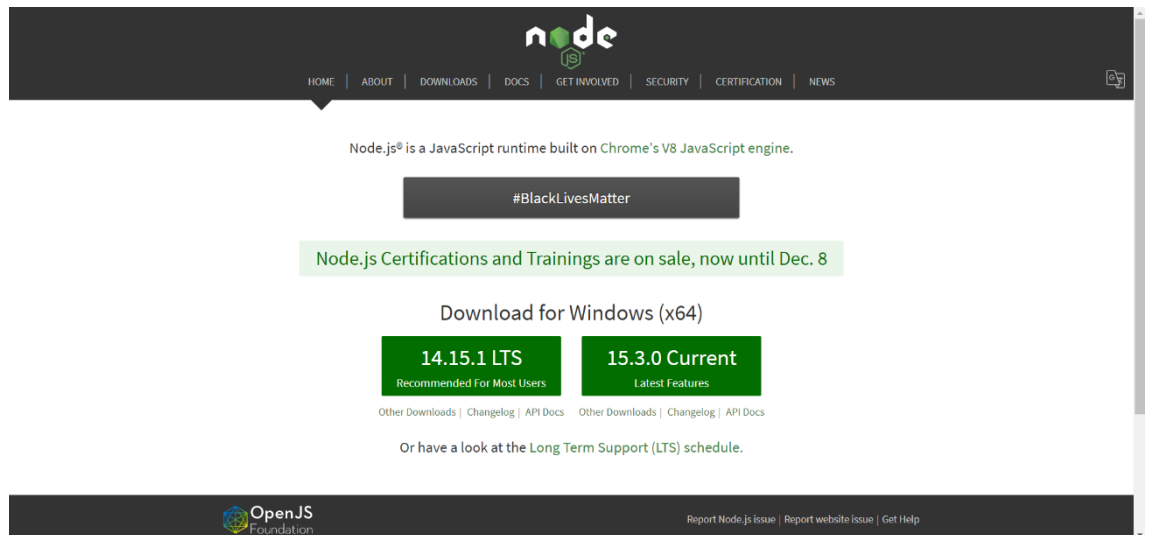
*Hình 3.5. Adapter 5V.*

- Điện áp đầu vào: AC100-240V 50/60HZ
- Điện áp ra: 5VDC
- Dòng điện ra: Max 1A
- Chiều dài đường dây đầu ra: tổng chiều dài 1,2m
- Đóng gói: Hộp trắng
- Ổ cắm DC: 5.5 \* 2.5mm

### **3.3. Cài đặt các nền tảng và cấu hình cần thiết**

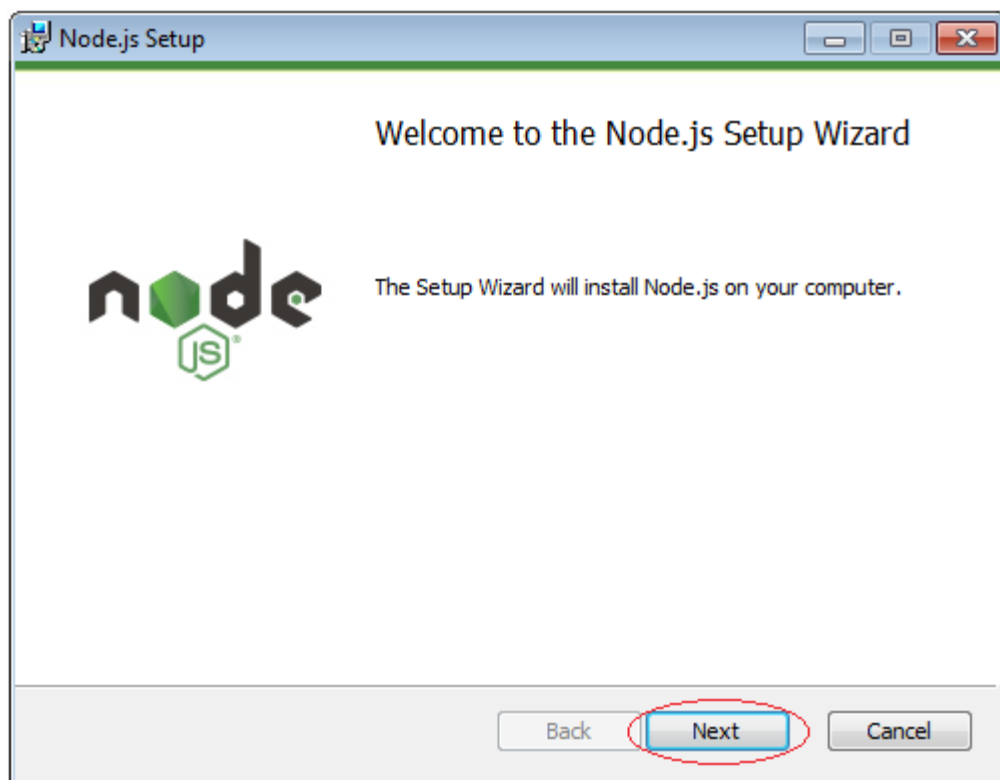
#### **3.3.1. Cài đặt Node JS để làm server cho hệ thống.**

B1: Truy cập vào đường link <https://nodejs.org/en/> và dowload phiên bản node js mới nhất.



Hình 3.6. Trang chủ Node JS.

B2: Giải nén file mới tải về và bấm NEXT, giữ nguyên các cấu hình mặc định.



Hình 3.7. Giải nén file Node JS cho Windows.

Lúc này hệ điều hành đã có các config chạy ứng dụng node js, để kiểm tra: nhấn windows + R search cmd và enter, trong cửa sổ cmd nhập lệnh node -v để kiểm tra version hiện tại.



```
C:\Windows\system32\cmd.exe
C:\Users\tcthi>node -v
v12.18.3
C:\Users\tcthi>
```

Hình 3.8. Kiểm tra version Node JS.

### 3.3.2. Arduino IDE và cấu hình cho chip ESP32

Hiểu một cách đơn giản, Arduino IDE là 1 phần mềm giúp chúng ta nạp code đã viết vào board mạch và thực thi ứng dụng. Arduino IDE là chữ viết tắt của Arduino Integrated Development Environment, một công cụ lập trình với các board mạch Arduino. Nó bao gồm các phần chính là Editor (trình soạn thảo văn bản, dùng để viết code), Debugger (công cụ giúp tìm kiếm và sửa lỗi phát sinh khi build chương trình), Compiler hoặc interpreter (công cụ giúp biên dịch code thành ngôn ngữ mà vi điều khiển có thể hiểu và thực thi code theo yêu cầu của người dùng).

Hiện nay, ngoài các board thuộc họ Arduino, thì Arduino IDE còn hỗ trợ lập trình với nhiều dòng vi điều khiển khác như ESP, ARM, PIC, ...

Cấu trúc một chương trình Arduino bao gồm 2 phần chính :

```
void setup()
{ Thực hiện việc thiết lập ban đầu cho các ứng dụng.  }
void loop()
{ Vòng lặp thực hiện chương trình.  }
```

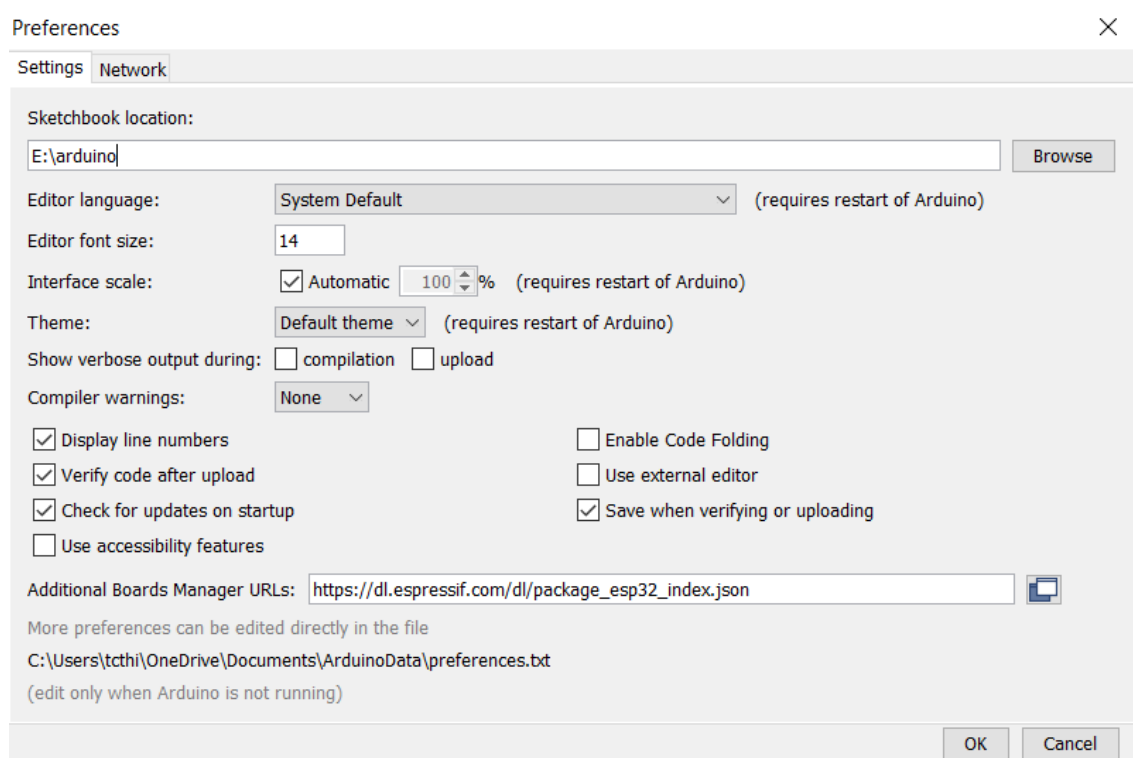
Hàm setup() được sử dụng để khởi tạo giá trị các biến, thiết lập chế độ chân, bắt đầu sử dụng các thư viện...Hàm setup chỉ thực hiện một lần khi cấp nguồn hoặc reset Arduino.

Hàm loop() được hiểu như là chương trình chính, thực hiện các chức năng được lập trình và có tính lặp lại liên tục.

**B1:** Cài đặt bộ công cụ Arduino IDE.

**B2:** File => Preferences, dưới phần Additional Boards Manager URLs thêm đường link [https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json) vào và bấm Ok.



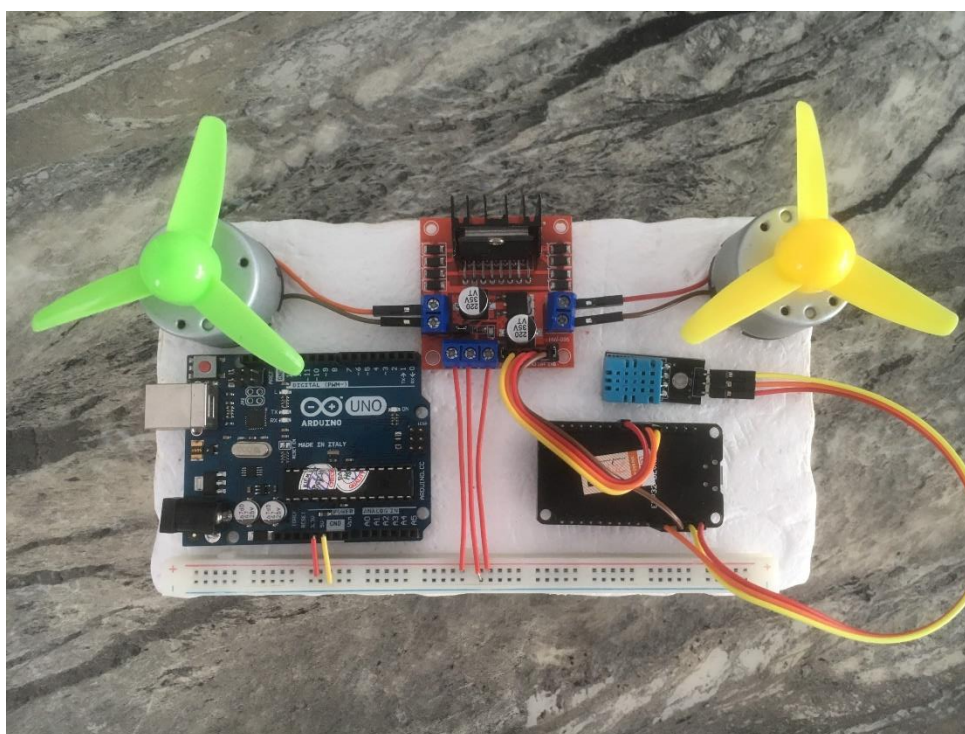


Hình 3.9. Thêm file thông tin cho ESP32.

### 3.4. Thiết kế hệ thống.

#### 3.4.1. Thiết kế phần cứng.

##### 3.4.1.1. Hệ thống mạch tổng quát.



Hình 3.10. Hệ thống mạch tổng quát.

### 3.4.1.2. Hệ thống mạch từng phần

#### Khối xử lý:

- Nguồn 3.3V: Kết nối chân VCC của module cảm biến DHT11.
- GND: Kết nối chân GND của module cảm biến DHT11.
- GPIO 4: Kết nối chân OUT của module cảm biến DHT11.
- GPIO 27: Kết nối chân IN1 module L298N.
- GPIO 26: Kết nối chân IN2 module L298N.
- GPIO 33: Kết nối chân IN3 module L298N.
- GPIO 15: Kết nối chân IN4 module L298N.

#### Khối nguồn:

- Nguồn 5V: Kết nối chân Vm, Vlogico của module L298N.
- GND: Kết nối chân GND của module L298N.

#### Khối cảm biến:

- VCC: Kết nối chân 3.3V của module ESP32.
- GND: Kết nối chân GND của module ESP32.
- OUT: Kết nối chân GPIO 4 của module ESP32.

#### Khối động cơ:

- 2 động cơ DC lần lượt kết nối vào 2 chân motor 1 và motor 2 của module L298N.

### 3.4.2. Thiết kế cơ sở dữ liệu.

Gồm các collection sau:

- *Sensors collection:*  
{  
    \_id: ObjectID,  
    name: String,  
    stationsID: ObjectID,  
}
- *Stations collection:*  
{  
    \_id: ObjectID,

```

        stationsID: ObjectID,
        name: String,
        coordinateID: ObjectID
    }

- Coordinates collection:
  {
    _id: ObjectID,
    lat: Double,
    long: Double
  }

- Measures collection:
  {
    _id: ObjectID,
    name: String,
    unitID: ObjectID,
    description: String
  }

- Thresholds collection:
  {
    _id: ObjectID,
    min: Double,
    max: Double,
    sensorID: ObjectID,
    measureID: ObjectID,
  }

- Units collection:
  {

```

```

        _id: ObjectID,
        name: String,
    }

-   Values collection:
    {
        _id: ObjectID,
        stationID: ObjectID,
        values: [
            {
                _id: ObjectID,
                sensorID: ObjectID,
                value: Double,
                measureID: ObjectID,
                date: DateTime
            }
        ]
    }

```

### 3.4.3. Thiết kế chức năng cho hệ thống.

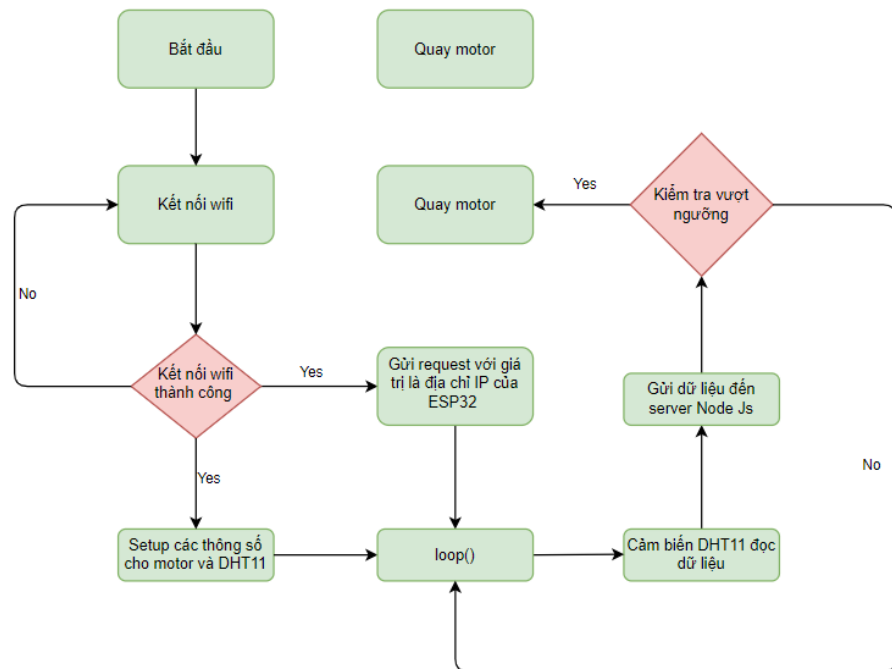
Đặc tả :

- Chức năng đăng nhập: Người dùng đăng nhập bằng tài khoản đã đăng ký hoặc bằng facebook.
- Chức năng xem thông tin User đã đăng nhập, sửa đổi các thông tin của User.
- Chức năng xem thông tin từng trạm.
- Chức năng xem dữ liệu realtime của từng trạm.
- Chức năng thống kê dữ liệu của tuần, tháng , năm.
- Chức năng xuất file excel.
- Chức năng cảnh báo bằng màu sắc khi quá ngưỡng cho phép.

- Chức năng cảnh báo bằng tin nhắn điện thoại khi quá ngưỡng cho phép.

### 3.4.4. Thiết kế phần mềm.

#### 3.4.4.1. Lưu đồ thuật toán của phần lập trình nhúng ESP32.



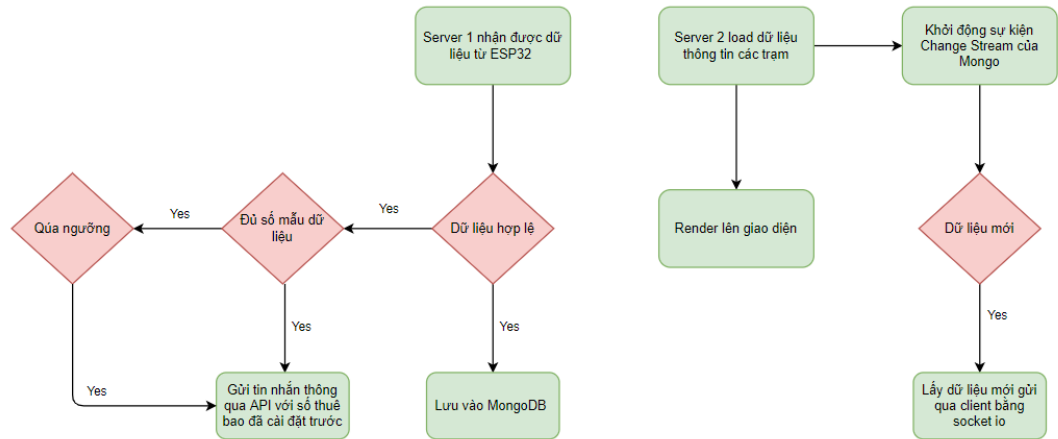
Hình 3.11. Lưu đồ thuật toán phần lập trình nhúng.

Giải thích lưu đồ:

- Bắt đầu ESP32 kết nối với wifi như đã cài đặt trong code, nếu thất bại thì kết nối lại, nếu thành công gửi địa chỉ IP mà ESP32 truy cập của mạng wifi hiện tại cho server node js (địa chỉ IP này server node sẽ dùng để gửi request cho ESP32).
- Khi wifi kết nối thành công, sẽ cài đặt tiếp các thông số, thư viện cho module L298N (điều khiển 2 motor), cảm biến DHT11.
- Vòng lặp loop được khởi động: cảm biến DHT11 bắt đầu đọc thông số nhiệt độ và độ ẩm, thông qua ESP32 gửi dữ liệu đến server node bằng wifi đã cài đặt trên phần setup, sau đó gửi dữ liệu đến server node js,

- Dựa vào dữ liệu kiểm tra ngưỡng, nếu vượt ngưỡng thì quay motor tương ứng, sau đó thực tiếp vòng lặp loop.

#### 3.4.4.2. Lưu đồ thuật toán phần Server.



Hình 3.12. Lưu đồ thuật toán phần Server.

Giải thích lưu đồ:

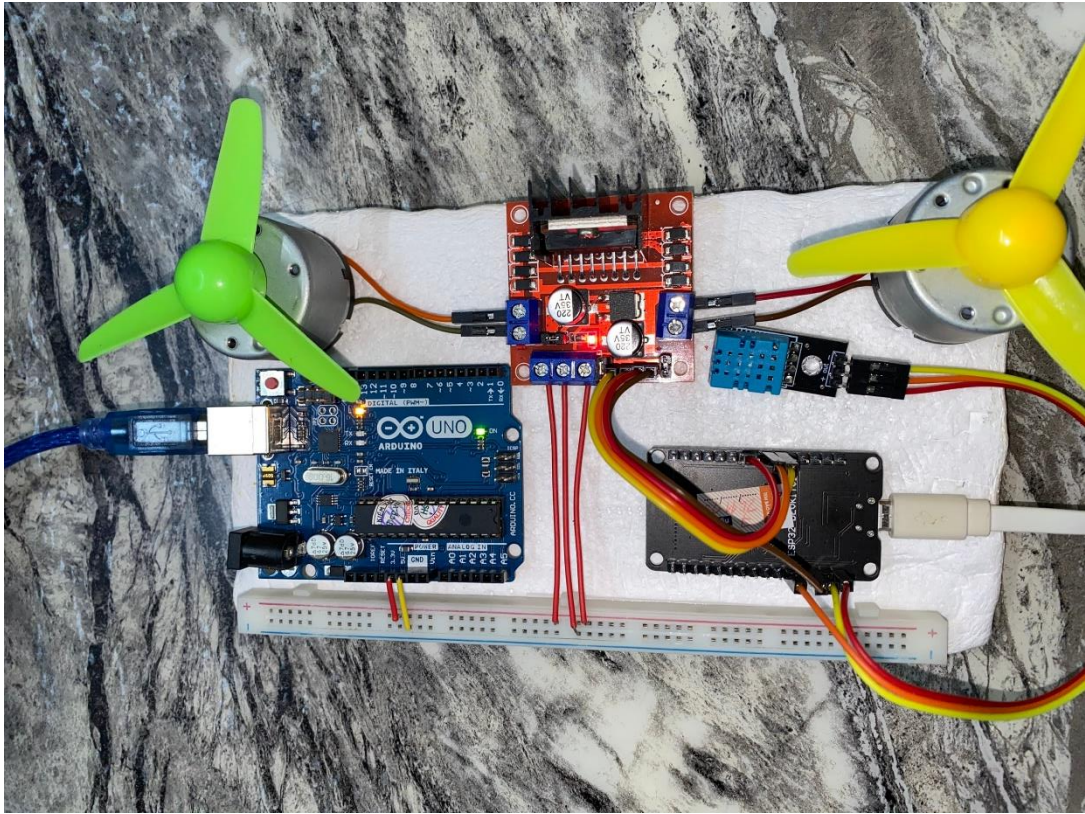
Hệ thống gồm 2 server: server 1 và server 2.

- Server 1 có các router để lắng nghe các request mang dữ liệu từ ESP32. Khi nhận được dữ liệu server sẽ tiến hành kiểm tra dữ liệu có hợp lệ hay không, nếu hợp lệ sẽ lưu vào database. Sau đó server 1 kiểm tra lần lượt nếu đủ số mẫu dữ liệu (mặc định 10) và dữ liệu bị quá ngưỡng sẽ gửi tin nhắn thông qua API cho thuê bao đã cài đặt trước.
- Server 2 khi khởi động sẽ load các dữ liệu như thông tin trạm, thông tin người dùng sẽ render ra giao diện, đồng thời khởi động sự kiện change stream (1 sự kiện lắng nghe sự thay đổi của database được cung cấp bởi mongoDB, lưu ý mongoDB phải được setup replica set mới có thể khởi động sự kiện này). Sự kiện change stream lắng nghe nếu có dữ liệu mới sẽ gửi qua client thông qua socket io (không render lại view).

## CHƯƠNG 4: KẾT QUẢ - ĐÁNH GIÁ - HƯỚNG PHÁT TRIỂN

### 4.1. Kết quả.

#### 4.1.1. Một số hình ảnh hệ thống phần cứng khi hoàn thành.




*Hình 4.1. Hệ thống phần cứng*


#### 4.1.2. Kết quả từng chức năng.


- Chức năng đăng ký: cho phép người dùng đăng ký tài khoản để sử dụng. Tài khoản sẽ dùng để đăng nhập với thông tin của người dùng.







## Sign up

 Your Name

 Your Email

 Password

 Repeat your password


☐ I agree all statements in Terms of service

[I am already member](#)


Register


Hình 4.2. Chức năng đăng ký tài khoản.

- Chức năng đăng nhập:



## Sign up




 Your Name

 .....

☐ Remember me

Log in

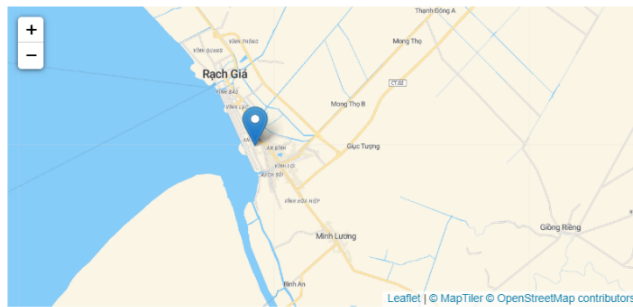
Create an account

Or login with   

Hình 4.3. Chức năng đăng nhập.

- Chức năng thêm trạm mới : cho phép người chọn tên trạm, vị trí của trạm, vị trí sẽ được API convert sang địa chỉ chi tiết





Tọa độ 9.967498 105.103477

Địa chỉ chi tiết Rach Giá, Kiên Giang

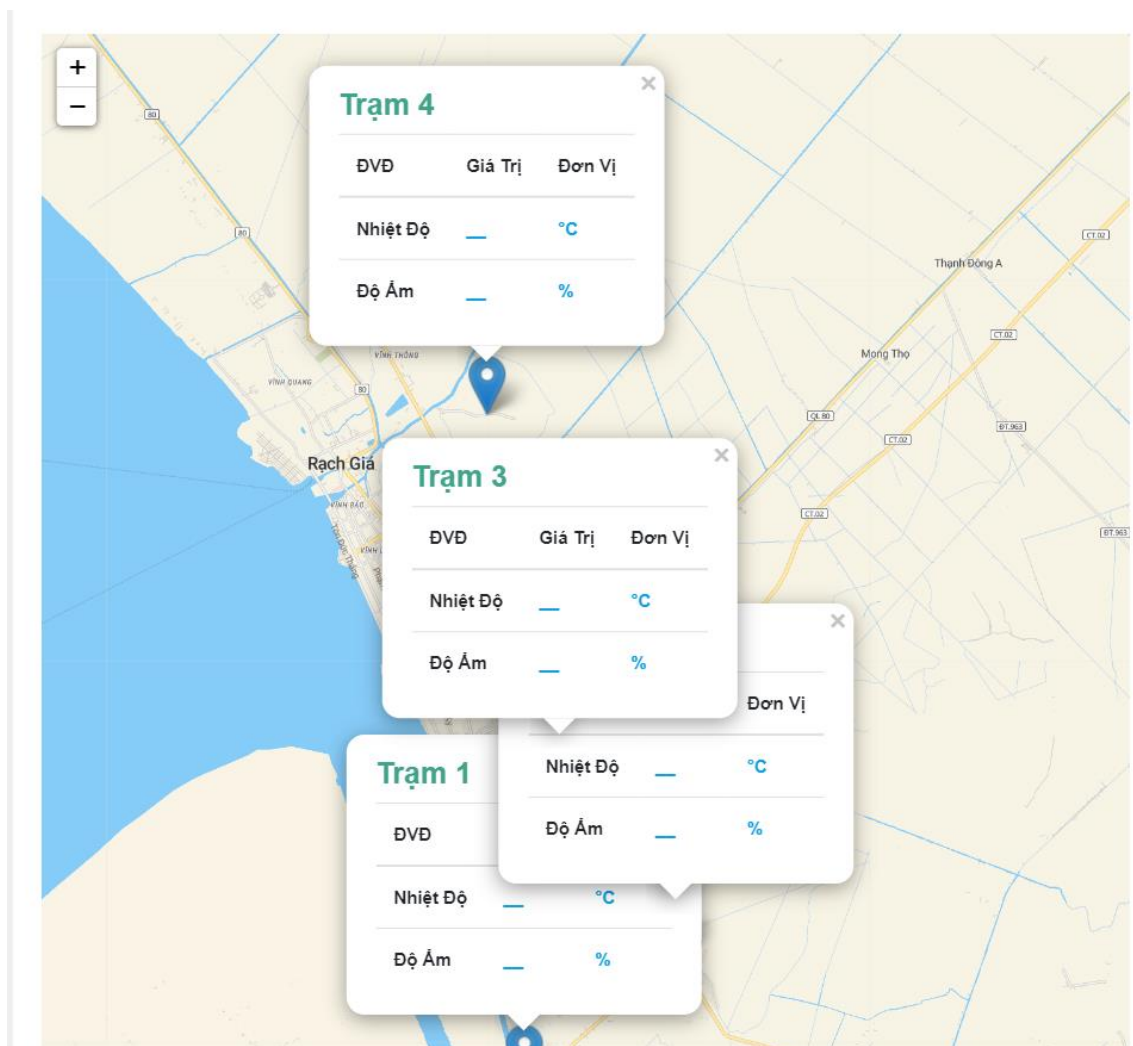
Tên trạm Station John

Thoát

Thêm trạm

Hình 4.4. Chức năng thêm trạm mới.

- Chức bản đồ hiển thị nhiều trạm: Hiển thị thông tin của nhiều trạm với vị trí cụ thể trên bản đồ, người dùng có thể xem nhiệt độ, độ ẩm hiện tại của trạm nhiều trạm cùng 1 lúc



Hình 4.5. Chức năng bản đồ xem nhiều trạm.

- Chức năng xem thông tin trạm:

Chi Tiết

×

Tên Trạm

Trạm 1

[ 9.872367, 105.125572 ]

Tọa Độ

Địa Chỉ

Đường Vào Phà Tắc Cậy Cũ, Xã Bình An, Huyện Châu Thành, Kiên Giang

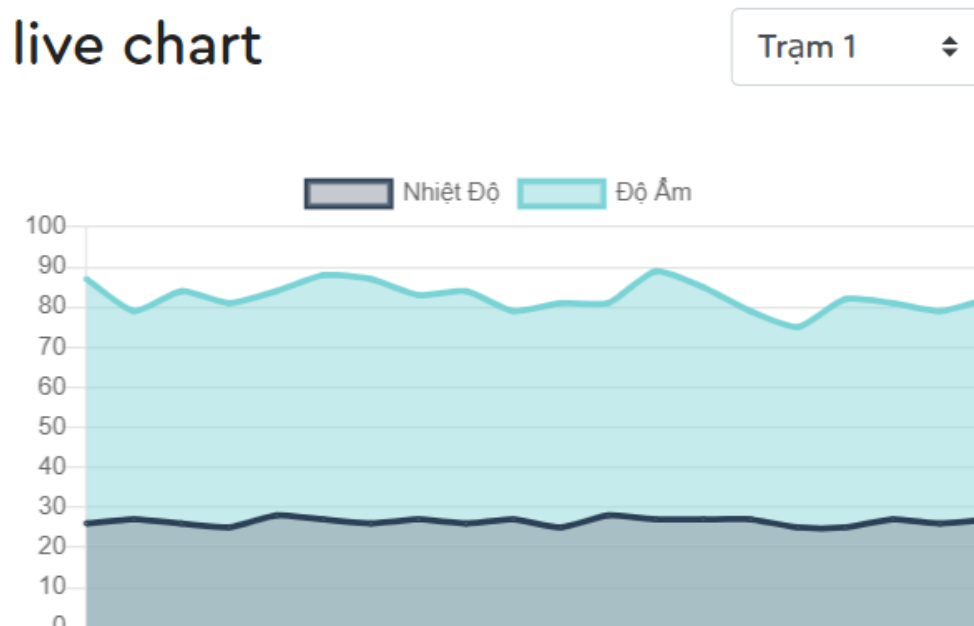
#	Cảm Biến	Ngưỡng Nhiệt Độ	Ngưỡng Độ Ẩm	Ngưỡng Độ Ẩm
1	dh11	[ 26 - 30 ]	[ 74 - 85 ]	

Close

Save changes

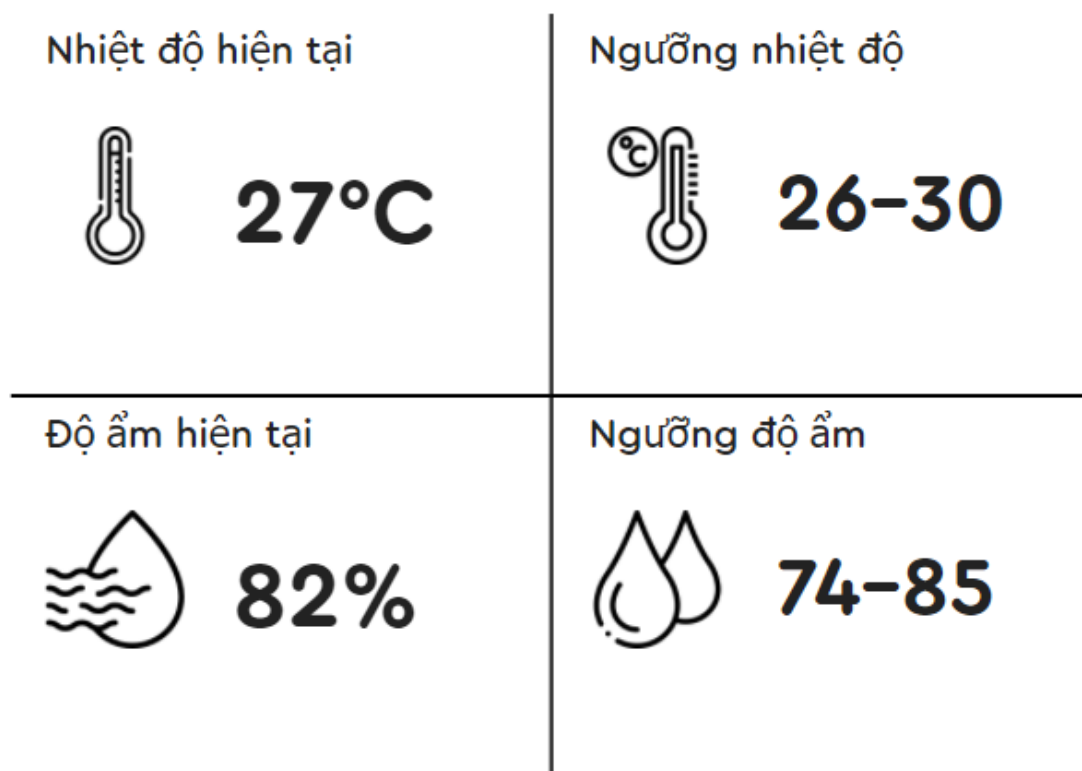
Hình 4.6. Chức năng xem thông tin trạm.

- Chức năng hiển thị dữ liệu dạng biểu đồ:



Hình 4.7. Chức năng xem dữ liệu realtime.

- Chức năng hiển thị dữ liệu hiện tại:



Hình 4.8. Chức năng hiển thị dữ liệu hiện tại

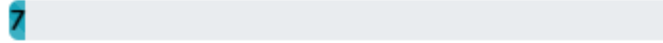
- Chức năng hiển thị thống kê dữ liệu:



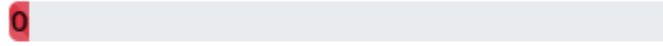
Hình 4.9. Chức năng thống kê dữ liệu.

- Chức năng thống kê số lần vượt ngưỡng.

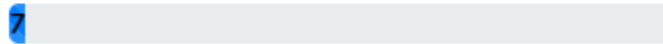
Nhiệt độ ( số lần )



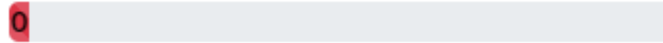
Vượt ngưỡng



Độ ẩm ( số lần )

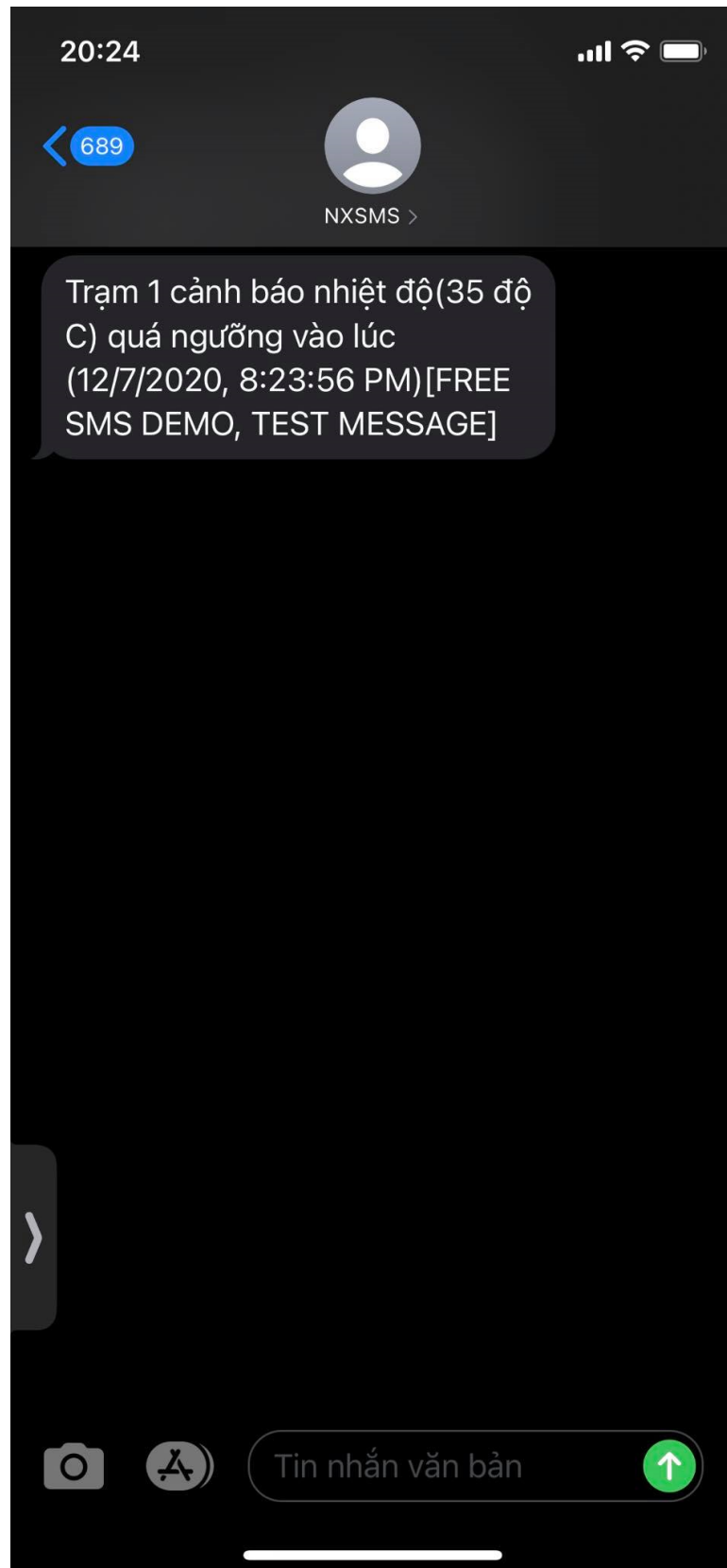


Vượt ngưỡng



*Hình 4.10. Chức năng thống kê số lần vượt ngưỡng.*

- Chức năng cảnh báo bằng tin nhắn điện thoại



Hình 4.11. Chức năng cảnh báo bằng tin nhắn điện thoại.

## 4.2. Đánh giá và hướng phát triển

*Ưu điểm:*

- Hệ thống chạy realtime khá tốt
- Giao diện thân thiện với người dùng dễ dàng sử dụng.

*Hạn chế:*

- Giao diện chưa thể tùy biến theo thiết bị truy cập (responsive)
- Hệ thống cảnh báo bằng điện thoại hoạt động chưa trơn tru vì phải sử dụng API.
- Chức năng đăng nhập chưa phân quyền.
- Mô hình hệ thống còn phức tạp.
- Phần cứng chưa tối ưu.

*Hướng phát triển:*

- Responsive giao diện để tương thích với mọi loại thiết bị
- Phân quyền chức năng đăng nhập.

## **TÀI LIỆU THAM KHẢO**

- [1] **Đỗ Trung Kiên**, Giáo trình Lập Trình Nhúng 2019.
- [2] **Marco Schwartz**, Internet of Things with ESP8266, Birmingham, 2018.